Riiid 데이터를 이용한 토익점수 예측

정현석, 정소비, 정연규

CONTENTS

토익공부현황

ETS TOEIC

토익 스트레스, 취업절망감 유발 우려

토익점수도 중요하지만 객관적인 실력 갖춰야

진짜 토익 때문에 스트레스 받아죽겠네요 ㅠㅠ

토익 스트레스때문에 헛구역질이...

토익공부만 하려면 스트레스 받고 짜증나요.

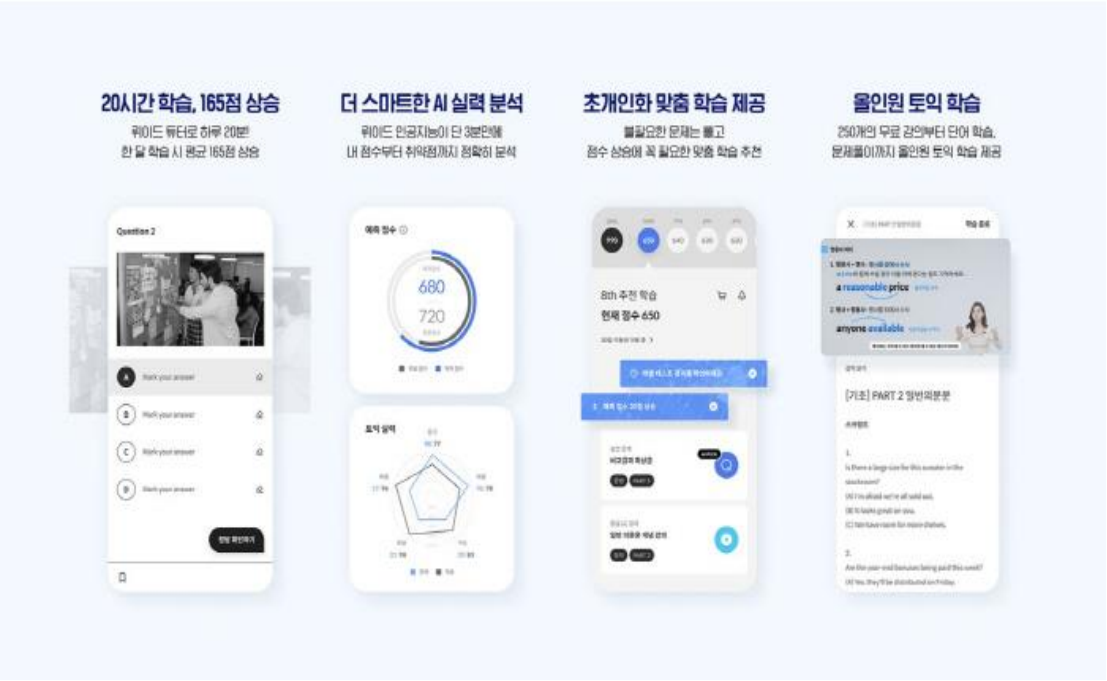
'소확행' 말고 '취확행'...핸드폰 반납하고 하루 14시간 토익공부

01

주제 선정 배경

AI 튜터 산타 토익/뤼이드튜터이란?

Riiid TUTOR



딥러닝 알고리즘을 기반으로 점수 예측 모델을 개발

학습행동을 분석

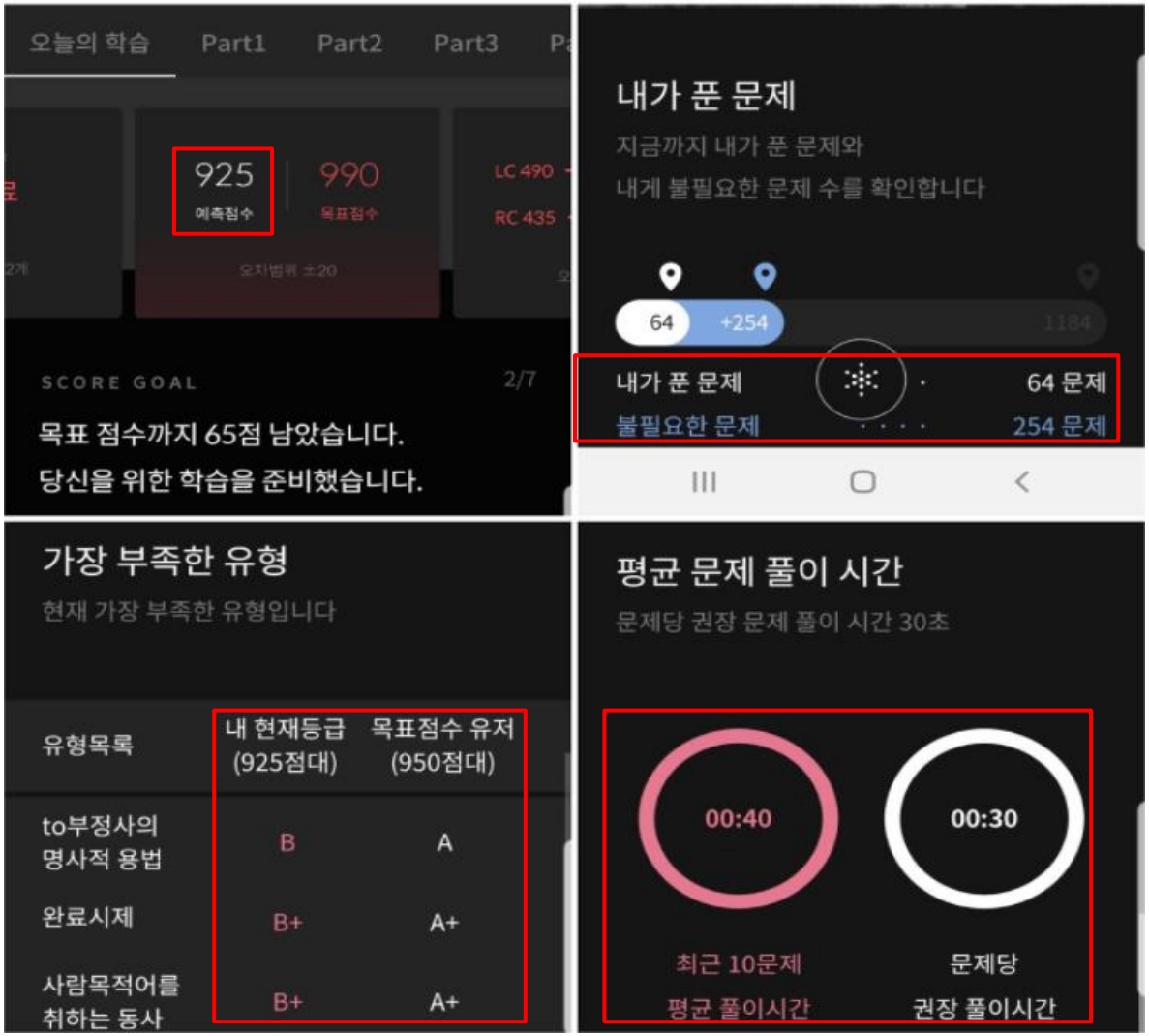
점수 예측 및 부족한 학습 추천

보다 효율적인 공부

주제 선정 배경

multicampus

Riiid 앱 방식



① 처음 모의고사를 풀면 예측 토익점수가 나옴

② 그 이후에 문제를 추천해 주는데 취약한 부분의 문제를 추천해줌

③ 문제를 풀 때마다 실시간으로 토익 점수를 예측해줌.

④ 평균 문제 풀이 시간을 제공하여 문제를 전략적으로 풀 수 있게 도와줌

⑤ 불필요한 문제를 제외하고 핵심적인 문제들만 추천해 줌

02

데이터 설명

train.csv
questions.csv
lectures.csv

Train.csv

train

	row_id	timestamp	user_id	content_id	content_type_id	task_container_id	user_answer	answered_correctly	prior_question_elapsed_time	prior_question_had_explanation
0	0	0	115	5692	0	1	3	1	NaN	NaN
1	1	56943	115	5716	0	2	2	1	37000.0	False
2	2	118363	115	128	0	0	0	1	55000.0	False
3	3	131167	115	7860	0	3	0	1	19000.0	False
4	4	137965	115	7922	0	4	1	1	11000.0	False
...	...	...	...	...	...	...	...	...	...	...
9999995	9999995	646228695	216747867	8805	0	410	1	1	23000.0	True
9999996	9999996	646321314	216747867	5708	0	411	0	0	13000.0	True
9999997	9999997	646393443	216747867	5434	0	412	0	0	66000.0	True
9999998	9999998	646445632	216747867	6429	0	413	3	1	37000.0	True
9999999	9999999	690225760	216747867	9010	0	414	2	1	16000.0	True

10000000 rows × 10 columns

row_id	***index***
timestamp	***사용자별 누적 학습 시간***
user_id	***사용자 고유 번호***
content_id	***id와 연결된 외래키***
content_type_id	***Content_id가 lectures와 연결된 값인지 questions와 연결된 값인지 구분하는 값 (0->문제, 1->강의)***

multicampus

Train.csv

train

	row_id	timestamp	user_id	content_id	content_type_id	task_container_id	user_answer	answered_correctly	prior_question_elapsed_time	prior_question_had_explanation
0	0	0	115	5692	0	1	3	1	NaN	NaN
1	1	56943	115	5716	0	2	2	1	37000.0	False
2	2	118363	115	128	0	0	0	1	55000.0	False
3	3	131167	115	7860	0	3	0	1	19000.0	False
4	4	137965	115	7922	0	4	1	1	11000.0	False
...	...	...	...	...	...	...	...	...	...	...
9999995	9999995	646228695	216747867	8805	0	410	1	1	23000.0	True
9999996	9999996	646321314	216747867	5708	0	411	0	0	13000.0	True
9999997	9999997	646393443	216747867	5434	0	412	0	0	66000.0	True
9999998	9999998	646445632	216747867	6429	0	413	3	1	37000.0	True
9999999	9999999	690225760	216747867	9010	0	414	2	1	16000.0	True

10000000 rows × 10 columns

task_container_id	***문제 타입 유형의 개수 (누적)***
user_answer	***사용자가 선택한 답 (0, 1, 2, 3), -1->강의***
answered_correctly	***사용자 선택한 답의 정답여부 (1->정답, 0->오답, -1->강의)***
prior_question_elapsed_time	***이전 문제를 푸는데 소요된 시간***
prior_question_had_explanation	***이전 문제/문제세트를 푼 후, 해설지 확인 -> True, 확인하지않음-> False***

Questions.csv / Lectures.csv

questions

	question_id	bundle_id	correct_answer	part	tags
0	0	0	0	1	51 131 162 38
1	1	1	1	1	131 36 81
2	2	2	0	1	131 101 162 92
3	3	3	0	1	131 149 162 29
4	4	4	3	1	131 5 162 38
...	...	...	...	...	...
13518	13518	13518	3	5	14
13519	13519	13519	3	5	8
13520	13520	13520	2	5	73
13521	13521	13521	0	5	125
13522	13522	13522	3	5	55

13523 rows × 5 columns

lectures

	lecture_id	tag	part	type_of
0	89	159	5	concept
1	100	70	1	concept
2	185	45	6	concept
3	192	79	5	solving question
4	317	156	5	solving question
...	...	...	...	...
413	32535	8	5	solving question
414	32570	113	3	solving question
415	32604	24	6	concept
416	32625	142	2	concept
417	32736	82	3	concept

418 rows × 4 columns

Question_id	*각각의 문제 번호*
Bundle_id	*문제/문제 묶음의 번호*
Correct_answer	*문제의 정답*
Part	*토익 Part. 1-7*
Tags	*비슷한 유형의 문제(들)*
Lecture_id	*각각의 강의번호* *(train 에 있는 content_id의 외래키)*
Tag	*강의를 위한 코드 넘버링*
Part	*토익 Part. 1-7*
Type of	*강의 종류* *(starter, intention, concept, solving question)*

03

탐색적 자료 분석 (칼럼 분석)

train.csv
questions.csv
lectures.csv

multicampus

Train.csv

```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000000 entries, 0 to 9999999
Data columns (total 10 columns):
 #   Column                          Dtype
---  ------                          -----
 0   row_id                          int64
 1   timestamp                       int64
 2   user_id                         int64
 3   content_id                      int64
 4   content_type_id                 int64
 5   task_container_id               int64
 6   user_answer                     int64
 7   answered_correctly              int64
 8   prior_question_elapsed_time     float64
 9   prior_question_had_explanation  object
dtypes: float64(1), int64(8), object(1)
memory usage: 762.9+ MB
```

```
train.user_id.nunique()
```

```
39491
```

총 사용자 수 : 39,491

```
train.content_id.nunique()
```

```
13749
```

이용한 강의/문제수:

13,749개

```
train.content_type_id.value_counts()
```

```
0    9804604
1     195396
Name: content_type_id, dtype: int64
```

문제를 푼 경우

9,804,604

강의를 본 경우

195,396

Train.csv

문제의 답

```
train.user_answer.unique()
```

```
array([ 3,  2,  0,  1, -1], dtype=int64)
```

사용자가 선택한 답

```
train.user_answer.value_counts().sort_index()
```

```
-1     195396
 0    2784028
 1    2660660
 2    1780388
 3    2579528
Name: user_answer, dtype: int64
```

Task_container_id

```
train[["user_id","task_container_id"]]
```

	user_id	task_container_id
0	115	1
1	115	2
2	115	0
3	115	3
4	115	4
5	115	5
6	115	6
38	115	38
39	115	39
40	115	40
41	115	40
42	115	40
43	115	41
44	115	41
45	115	41
46	124	0
47	124	1
48	124	2
49	124	3
50	124	4
...	...	...
9999995	216747867	410
9999996	216747867	411
9999997	216747867	412
9999998	216747867	413
9999999	216747867	414

10000000 rows × 2 columns

Train.csv

```
train.answered_correctly.value_counts()
```

```
 1    6457425
 0    3347179
-1     195396
Name: answered_correctly, dtype: int64
```

Answered_correctly

사용자가 선택한 답이 정답인 경우가 오답인 경우보다 약 2배가량 많음

```
train.prior_question_elapsed_time
```

```
0              NaN
1          37000.0
2          55000.0
3          19000.0
4          11000.0
             ...
9999995    23000.0
9999996    13000.0
9999997    66000.0
9999998    37000.0
9999999    16000.0
```

Prior_question_elapsed_time

이전 문제/문제세트를 푸는데 소요된 시간 ms단위이므로 약 10초에서 60초까지 다양하게 분포

```
train.prior_question_had_explanation.value_counts()
```

```
True     8855555
False    1105057
Name: prior_question_had_explanation, dtype: int64
```

Prior_question_had_explanation

이전의 문제/문제세트를 푼 후 해설을 확인한 경우: 8,855,555회,
확인하지 않은 경우: 105,057회

Questions.csv

```
questions.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13523 entries, 0 to 13522
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   question_id     13523 non-null  int64
 1   bundle_id       13523 non-null  int64
 2   correct_answer  13523 non-null  int64
 3   part            13523 non-null  int64
 4   tags            13522 non-null  object
```

Prior_question_had_explanation

```
questions.part.unique()
```

```
array([1, 2, 3, 4, 5, 6, 7]
```

토익 파트 : 1~7

Question_id

```
questions.question_id.nunique()
```

```
13523
```

문제수 : 13523개

Question_bundle

```
questions.bundle_id.nunique()
```

```
9765
```

문제 세트의 수 : 9765 개

Part

```
questions.part.value_counts()
```

```
1     992
2    1647
3    1562
4    1439
5    5511
6    1212
7    1160
```

파트별 문제의 수

Lectures.csv

```
lectures.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 4 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   lecture_id  418 non-null    int64
 1   tag         418 non-null    int64
 2   part        418 non-null    int64
 3   type_of     418 non-null    object
dtypes: int64(3), object(1)
```

강의 수

```
lectures.lecture_id.nunique()
```

```
418
```

강의 종류

```
lectures.type_of.value_counts()
```

```
concept             222
solving question    186
intention             7
starter               3
```

파트별 강의수

```
lectures.part.value_counts()
```

```
5    143
6     83
2     56
1     54
7     32
4     31
3     19
```

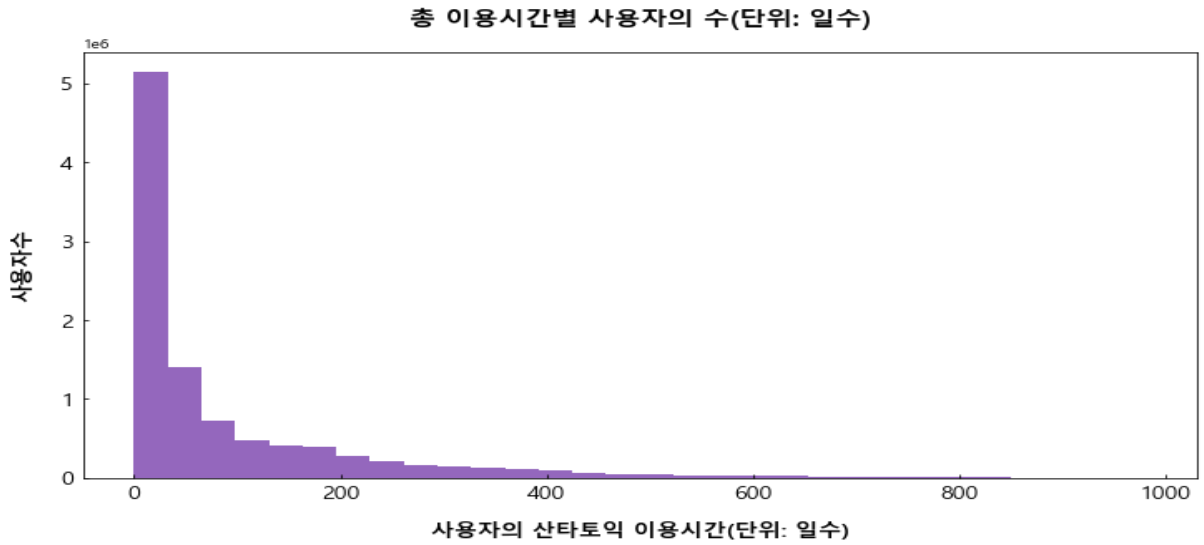
04

탐색적 자료 분석 (데이터 분포)

데이터 분포

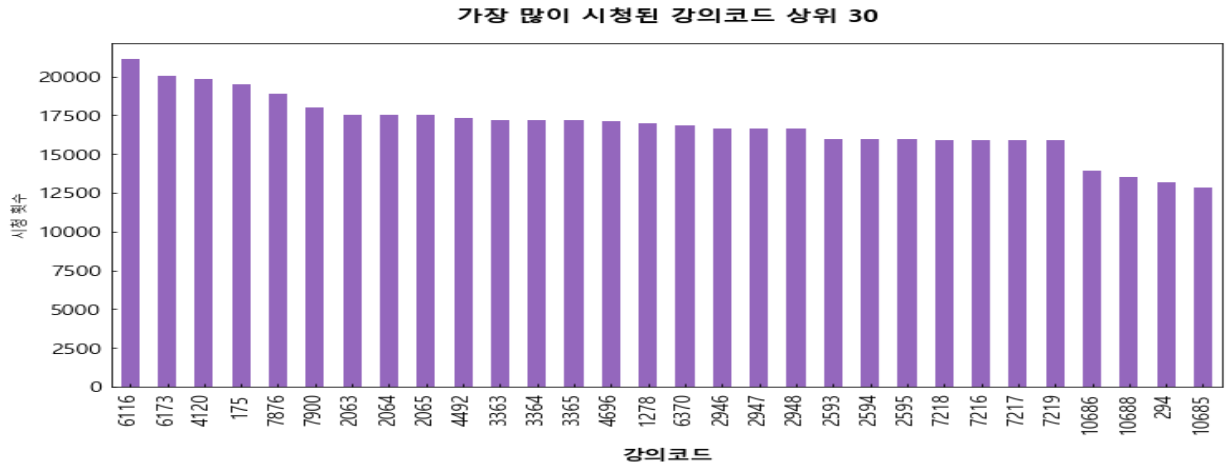
multicampus

train - timestamp / 각각 유저의 일수



Timestamp :
사용자별 학습 누적 시간
단위 ms => 일로 환산
(1 day = 24 * 60 * 60 * 1000 ms)

train - content_id / 갯수



content_id :
강의/문제 코드

value_counts와 slicing 적용

데이터 분포

multicampus

train - user_answer / 유저가 선택한 정답 비율

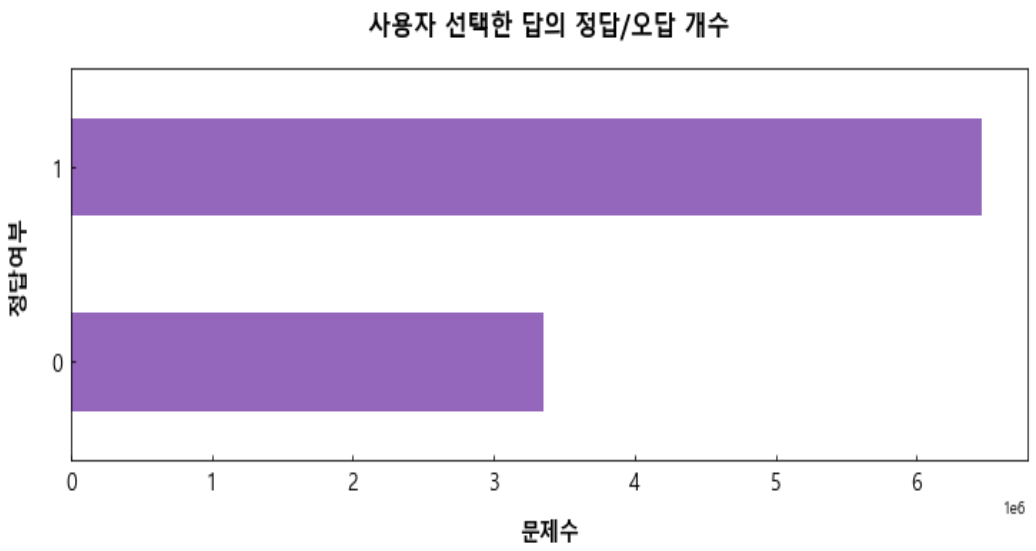
```
train[train.answered_correctly != -1].groupby('answered_correctly').size()
# train.pivot()
```

```
answered_correctly
0    3347179
1    6457425
dtype: int64
```

0(오답) : 3,347,179개 (34.1%)
1(정답) : 6,457,524개 (65.9%)

총 응답수 : 9,804,703개

train - user_answer / 유저가 선택한 정답 비율



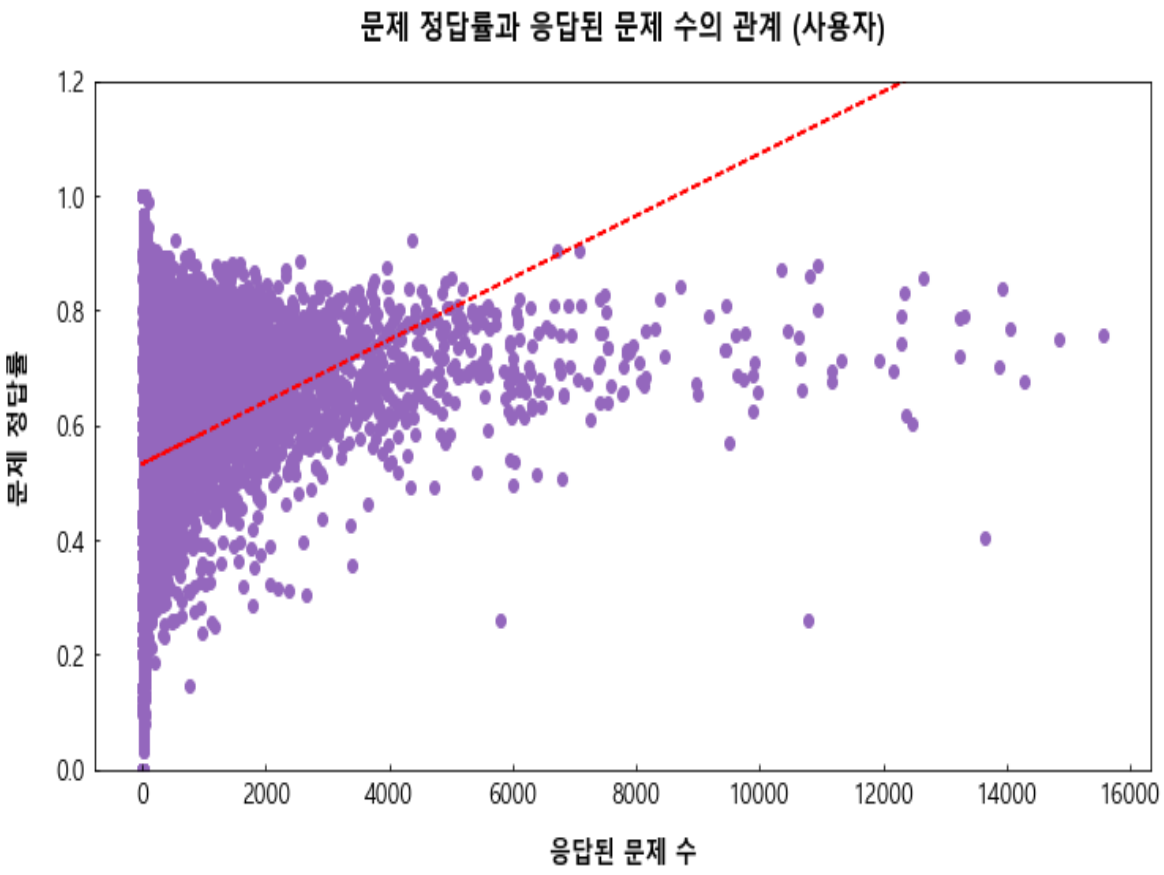
answered_correctly:
유저가 선택한 답의 정답 여부
(element 중 -1 값은 강의를 봤을 경우에 해당되므로 이를 제외하고 나타냄)

정답이 오답의 2배

데이터 분포

multicampus

train - user_percent / 유저가 문제푼 수와 실제 맞은 비율 간의 관계



user_id	Mean	Answers
115	0.695652	46
124	0.233333	30
2746	0.578947	19
5382	0.672000	125
8623	0.642202	109
8701	0.588235	17
12741	0.573585	265
13134	0.706356	1243
24418	0.690275	6283
24600	0.340000	50
32421	0.466667	30
40828	0.630435	92
44331	0.587629	291
45001	0.233333	30
46886	0.613636	44

x=user_percent.Answers / y = user_percent.Mean
x, y 간의 산점도 그래프
빨간색 선은 polyfit, poly1d를 이용한 x,y의 선형식

각 사용자의 문제 푼 수와 정답률 예시
(Mean = 정답 / (정답 + 오답))

데이터 분포

multicampus

train - prior_question_elapsed_time / 하나 푸는 데 걸리는 평균 시간(초)

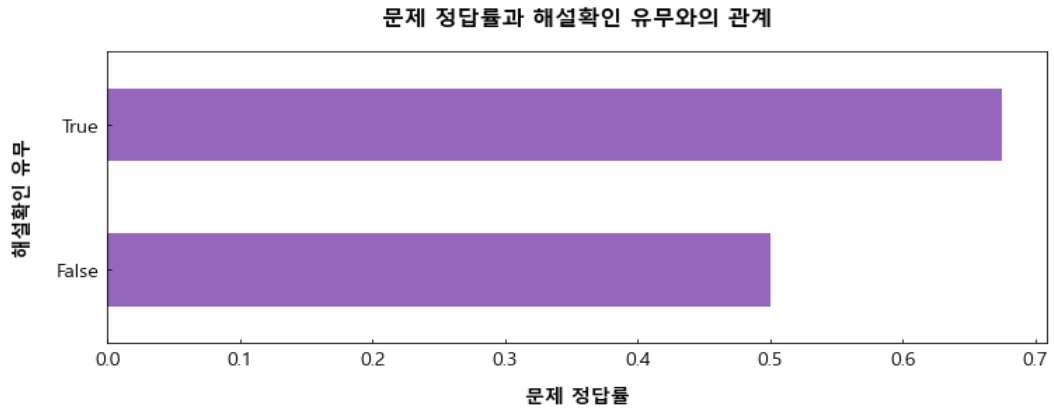
answered_correctly	answered_correctly	prior_question_elapsed_time
0	3347179	25704.232108
1	6457425	25357.034974

prior_question_had_explanation	answered_correctly mean	answered_correctly count
False	0.500357	909661
True	0.674771	8855555
NaN	0.680309	39388

answered_correctly : 정답/오답,
prior_question_elapsed_time : 문제를 푸는데 소요된 시간

분석 => 정답을 맞힌 문제와 틀린 문제를 푸는데 소요된 시간 시간은 큰 차이가 없음

train - prior_question_had_explanation 해설지보고 맞은 비율



해설지 확인 여부에 따른 정답 비율간의 관계 확인 (NaN은 강의를 봤을 경우에 해당)

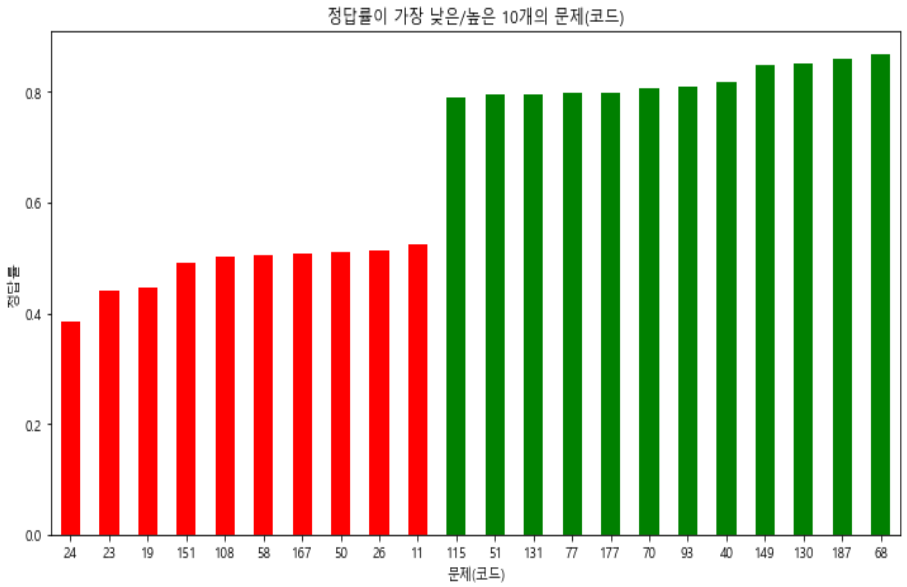
사용자는 문제를 푼 뒤 해설지를 확인하였으며, 해설지를 확인한 사용자가 정답 비율이 더 높은 걸 알 수 있다.

데이터 분포

multicampus

question - tags_select / 어려운태그, 쉬운태그 탑 10

tag	Wrong	Right	Total_questions	Percent_correct
24	15523	9653	25176	0.383421
23	25673	20228	45901	0.440688
19	17267	13833	31100	0.444791
151	26954	25882	52836	0.489855
108	23213	23511	46724	0.503189
...	...	...	...	...
40	4134	18343	22477	0.816079
149	5939	33315	39254	0.848703
130	3605	20508	24113	0.850496
187	3470	21414	24884	0.860553
68	1583	10310	11893	0.866896



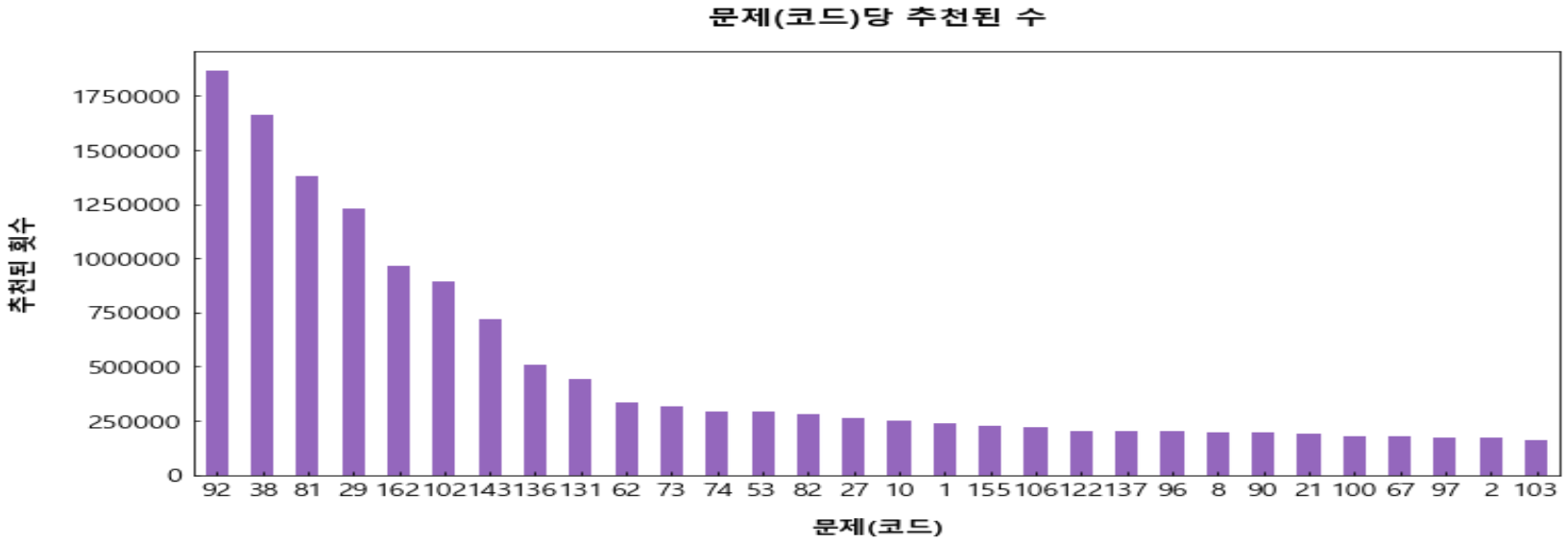
tag당 문제 정답률을 나타낸 데이터프레임
Wrong : 태크당 틀린 총 갯수
Right : 태크당 맞은 총 갯수
Total_questions : 총 문제 수
Percent_correct : 맞은 비율

정답률이 가장 낮은/높은 10개의 문제 (코드) 시각화
color = red : 정답률 낮음
color = green : 정답률 높음

데이터 분포

multicampus

question - tags_select / 태그당 사용된 횟수



문제(코드)당 추천된 수

tags = 한 tag의 비슷한 tag들의 다발
즉, 다른 tag와 가장 교집합이 많은 것을 알아보는 작업

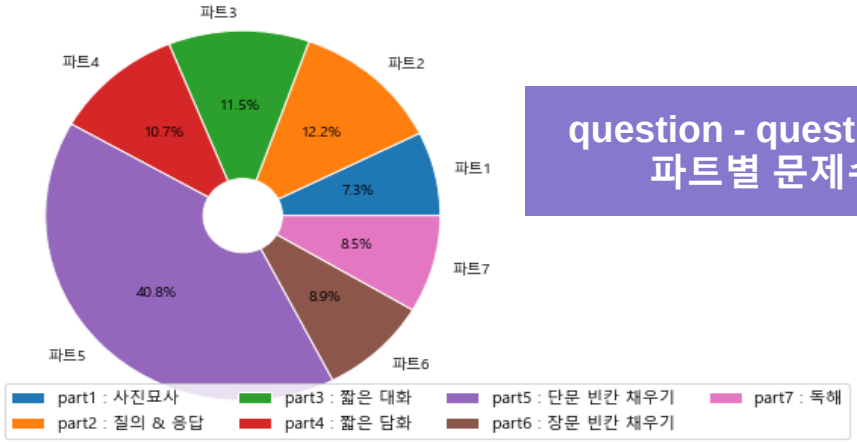
상위 8~9개의 문제 코드는 다른 코드들에 비해 상당히 많이 추천됨

데이터 분포

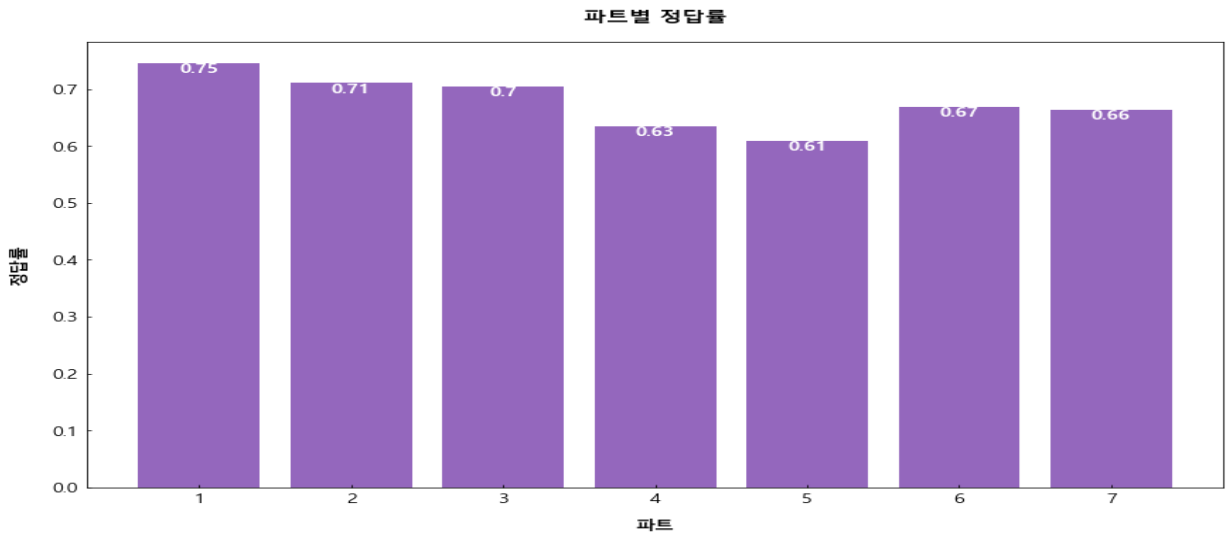
multicampus

파트별 문제수 비율 (pie)

part1 :사진묘사
part2 : 질의 & 응답
part3 : 짧은 대화
part4 : 짧은 담화
part5 : 단문 빈칸 채우기
part6 : 장문 빈칸 채우기
part7 : 독해



question - question_id
파트별 문제수

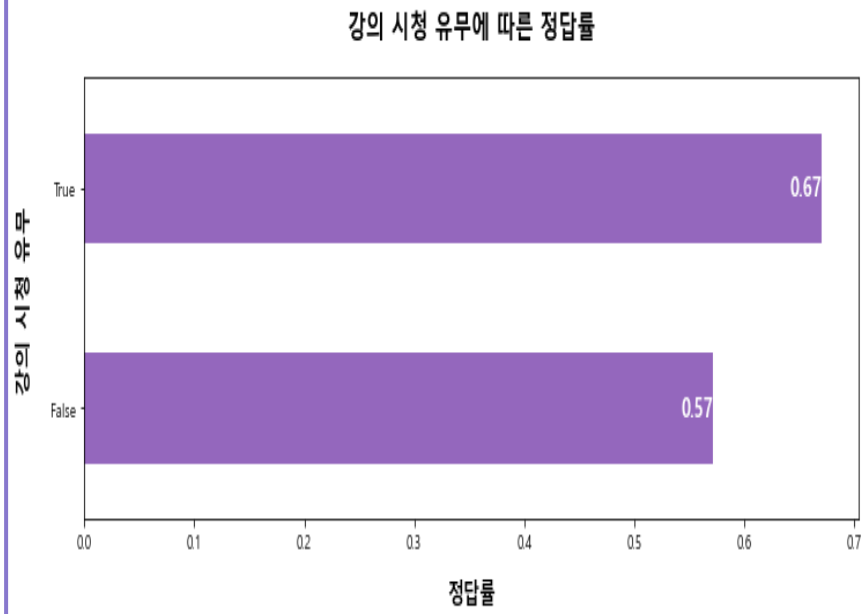


question- part
파트별 정답률

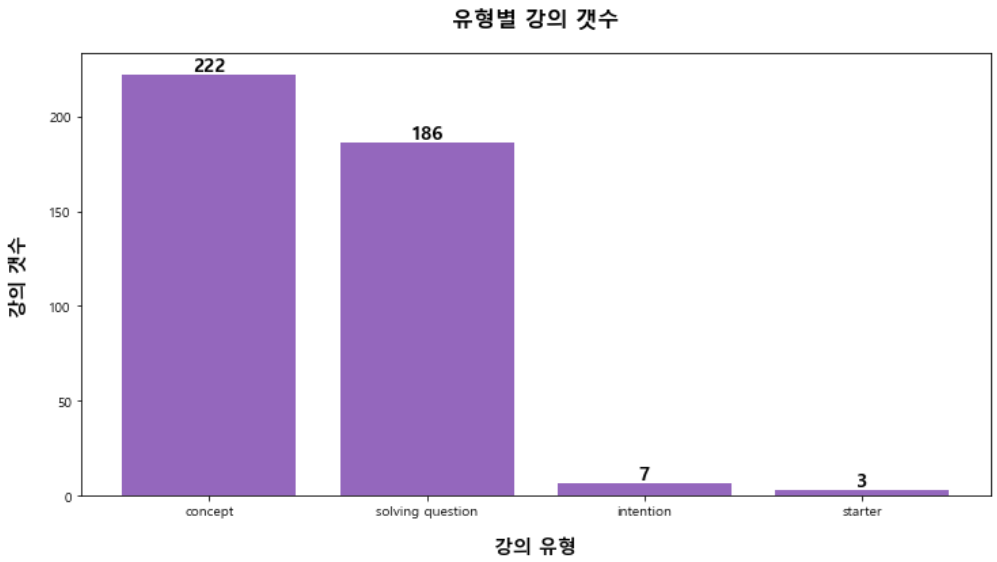
데이터 분포

multicampus

train - user_id,answered_correctly / 강의시청 유무별 정답률



lecture - type_of / 강의 유형별 개수



강의 시청 유무에 따른 정답률

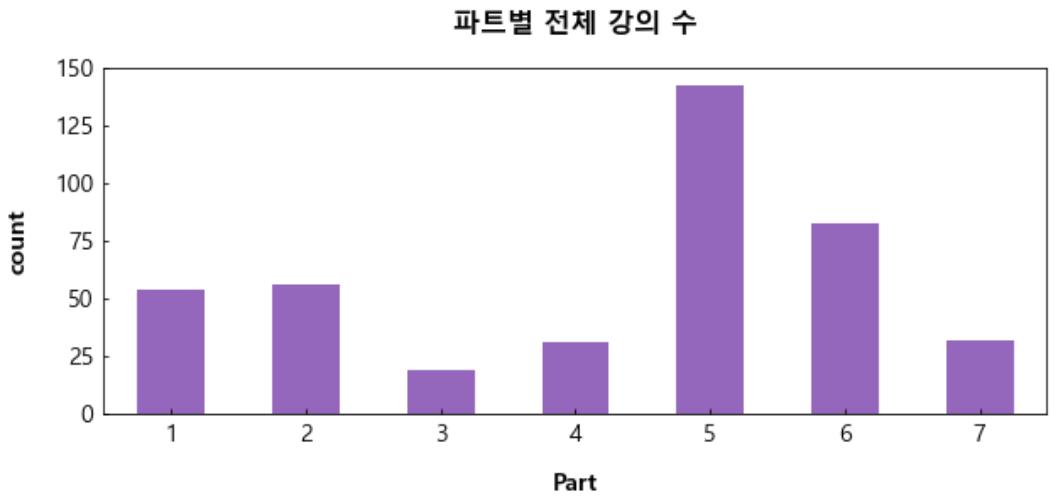
문제 관련 강의를 본 후 정답률은 그렇지 않은 집단보다 더 높다. 이는 강의를 시청하는 것이 문제를 맞히는데 효과적인 것을 알 수 있음

강의 유형별 개수

concept /solving question /intention / starter

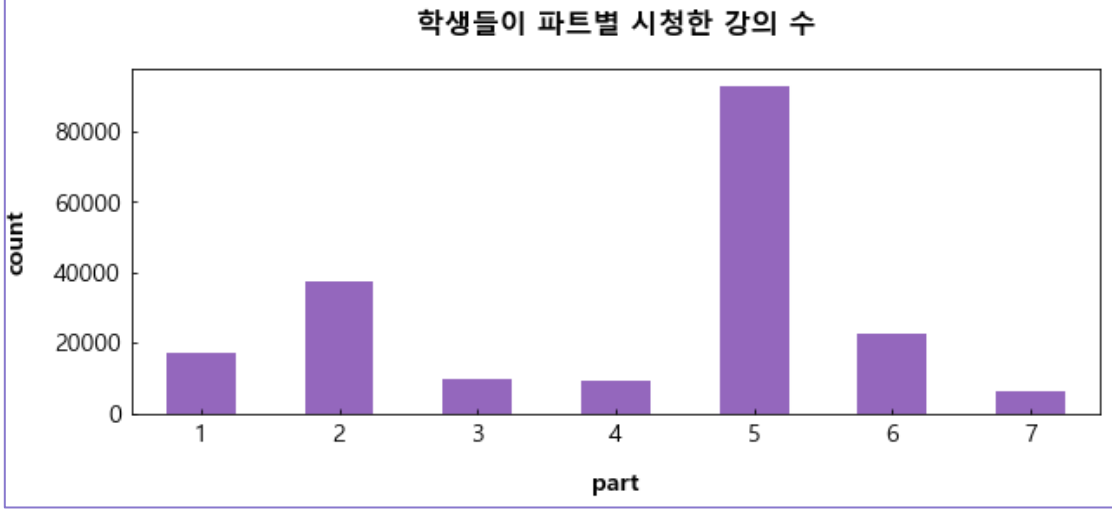
데이터 분포

multicampus



lecture - type_of,lecture_id
강의 파트별 갯수

groupby - part
count() -> lecture_id로 표현



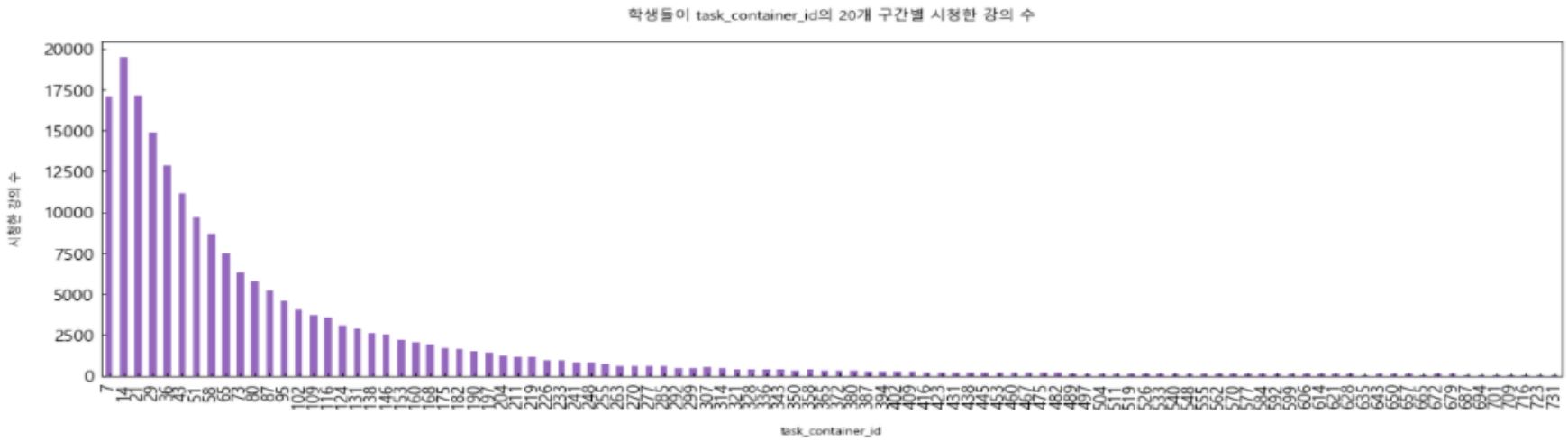
학생들이 파트별
시청한 강의수

파트5인 문법 부분에서 많이 취약한 것을 알 수 있으며, 듣기부분에서는 질의/응답에서 많이 틀려 강의를 찾는 것으로 파악된다.

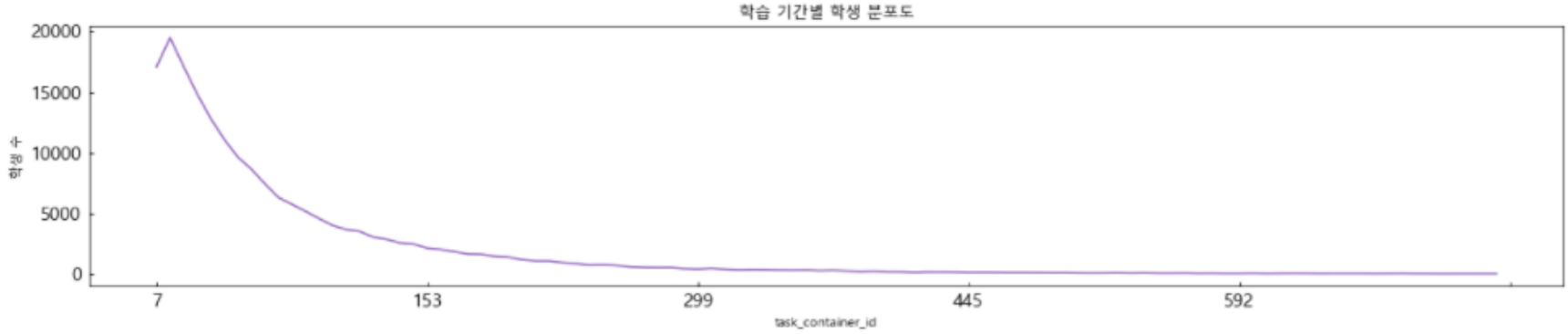
데이터 분포

multicampus

train_lecture - task_container_id, user_id / 학생들이 task_container_id의 20개 구간별 시청한 강의



train_lecture - task_container_id, user_id / task_container_id 의 최댓값별 학생

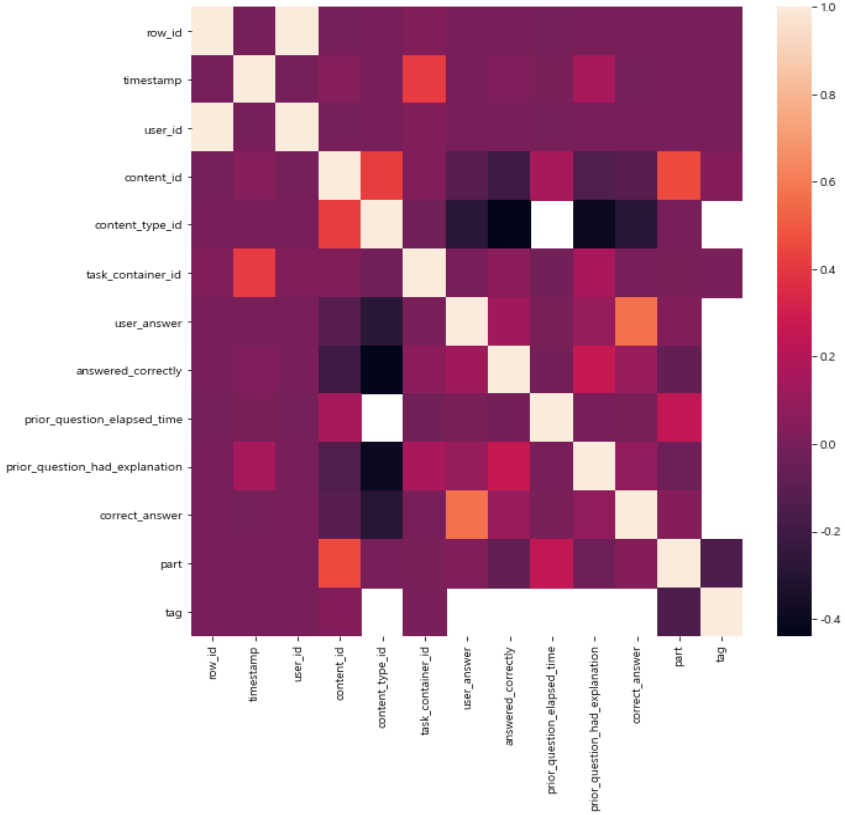


05

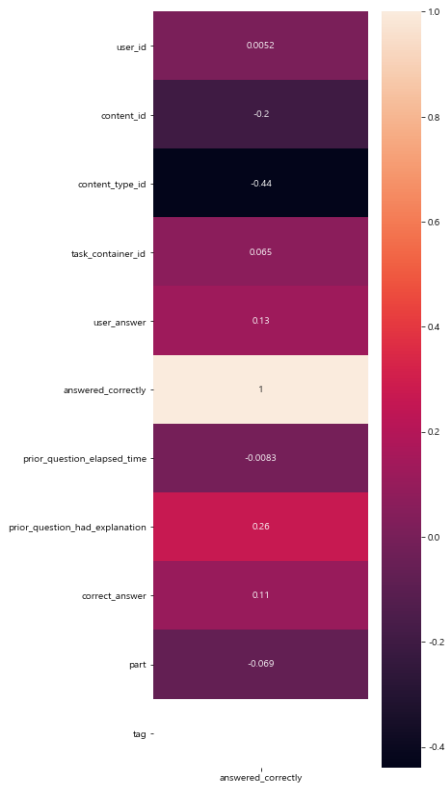
모델 검증

multicampus

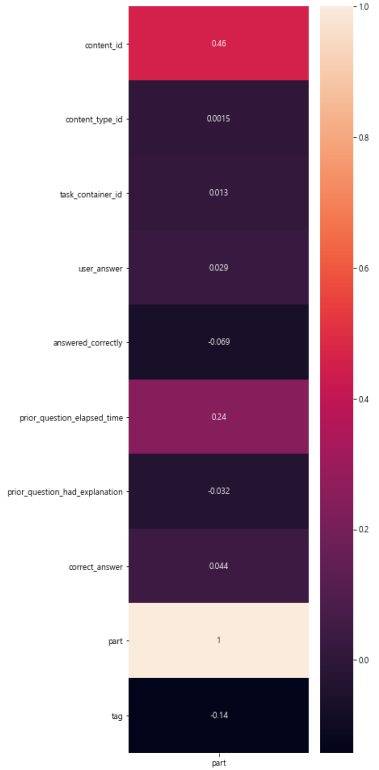
correlation analysis



total correlation analysis



answer_correctly : corr()



part : corr()

모델 검증

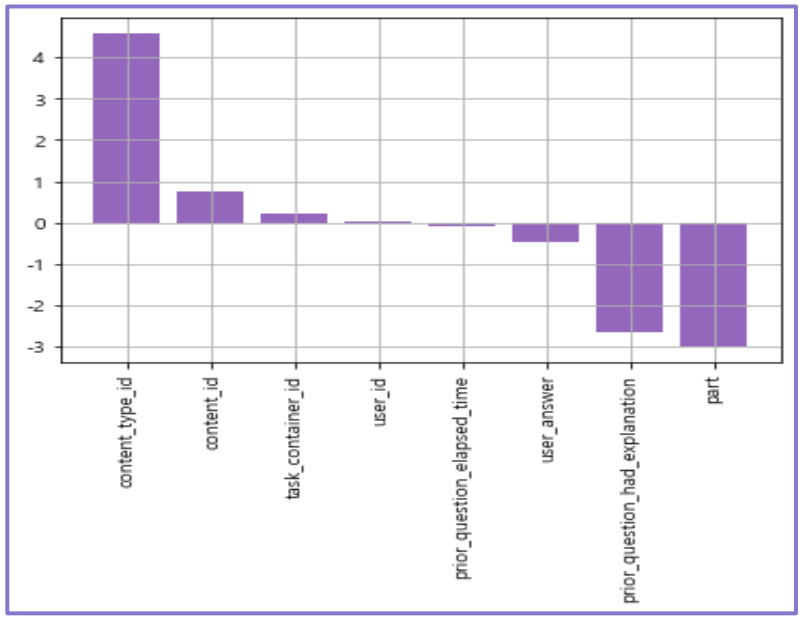
multicampus

	precision	recall	f1-score	support
-1	1.00	1.00	1.00	58635
0	0.53	0.08	0.14	1003574
1	0.67	0.96	0.79	1937791
accuracy			0.67	3000000
macro avg	0.73	0.68	0.64	3000000
weighted avg	0.63	0.67	0.58	3000000

사용 모듈 및 작업 sklearn.processing의 StandardScaler로 표준화

sklearn.model_selection 의 train_test_split로 train과 test 분리 (0.7 : 0.3)

sklearn.metrics에서 classification_report로 결과 보고서 출력



accuracy의 f1-score가 0.67로 상당히 좋은 결과 도출.

그래프는 Logistic Regression 모델의 coef_ 속성을 plot

06

한계점 및 느낀점

한계점

1. 데이터 용량이 큼
- 데이터 용량이 커 10**7의 데이터로 분석

1. 데이터 전처리
- 전처리 작업을 했지만, 상관 분석을 할 때, 좋지 않은 결과가 나왔다. 하지만 Accuracy가 0.67로 높게 나왔다

1. 딥러닝 기반 알고리즘
- EDA와 모델 검증까지만 했지만, 딥러닝 기반 알고리즘을 이용한다면 더욱 높은 정확도를 얻을 것이다.

느낀점

1. 정연규 (편집자)
- 프로젝트에 들어가기 앞서 데이터 유무의 중요성을 깨달았고, 공부 해야 할 부분이 어떤 부분인지 알 수 있는 프로젝트였다.

1. 정소비 (발표자)
- 분석에 적합한 데이터와 그렇지 않은 데이터에 대해 알 수 있었고, 데이터 분석과 시각화에 대해 많이 공부할 수 있는 시간이었다.

1. 정현석 (조장)
- 모르는 것들을 직접 부딪히면서 습득한 것이 좋았다. 하지만 아직도 배워야 할 게 많고 부족한 부분을 많이 느꼈다.

참고자료 / 분석도구

참고문헌

기사(토익 스트레스, 취업절망감 유발 우려)

http://www.ohmynews.com/NWS_Web/View/at_pg.aspx?CNTN_CD=A0000093981

기사('소확행'말고 '취확행'...핸드폰 반납하고 하루 14시간 토익공부)

https://gall.dcinside.com/board/view/?id=English&no=350988

기사(토익 스트레스, 취업절망감 유발 우려)

http://www.ohmynews.com/NWS_Web/View/at_pg.aspx?CNTN_CD=A0000093981

Riiid 홈페이지 : https://riiid.com/ko/tech

뉴스기사(AI 토익 튜터 산타, '뤼이드 튜터'로 전면 개편)

https://www.mk.co.kr/news/business/view/2021/06/622792/

분석도구

Python(Jupeter Notebook, Scikit-learn ,Matplotlib ,Pandas ,Numpy), Excel

문서 작업 : google documents

코딩 작업 : github

파일 공유 및 대화 : Slack

시각화 작업 : Matplotlib, Seaborn

활용데이터

https://www.kaggle.com/c/riiid-test-answer-prediction

감사합니다.